

Policy Management across Multiple Platforms and Application Domains

Matthew Johnson, Jeffrey M. Bradshaw, Hyuckchul Jung, Niranjan Suri, and Marco Carvalho
Florida Institute for Human and Machine Cognition (IHMC)
{mjohnson, jbradshaw, hjung, nsuri, mcarvalho}@ihmc.us

Abstract

One of the challenges of building a policy management framework is making it flexible enough to handle differences in both policy semantics and enforcement strategies across multiple platforms and application domains. The system must be expressive enough in each application domain to provide the richness needed for interesting policies. It must also provide a simple and flexible enforcement mechanism for adaptation to a variety of systems. In this paper we discuss the application of the KAoS policy services framework to human-robot teamwork—an application that involves a variety of application domains and enforcement at different levels of control; from low level network resource control to high level organizational constraints and coordination management. The study culminated in an outdoor field exercise that required coordination of mixed sub teams composed of two people and five robots whose task was to find and apprehend an intruder on a Navy pier.

1. Introduction

One of the challenges of building a policy management framework is making it flexible enough to handle differences in both policy semantics and enforcement strategies across multiple platforms and application domains. Some policy management frameworks have evolved from a specialized focus on specific domain of application, like network management [1], governing of autonomous behavior [1], or access control [3][4]. However, for a policy-based system to be effective in multiple application domains, it must develop beyond such a specialized focus to handle a richer policy semantics (ideally, an easily extensible semantics) and a variety of enforcement strategies. For example, administration of access control requires the specification of who can or cannot use specific resources, while an application to human-robot teamwork domain might require the specification of a policy to notify your teammates if you fail in the performance of a task. Available enforcement strategies and mechanisms also vary across platforms and application domains, and the

policy framework must be flexible enough to adapt to each of them. The policy system must also deal with the particular idiosyncrasies of the environment in which the application will be deployed. For example, some operating environments afford virtually guaranteed connectivity with ample network bandwidth, while other environments suffer from intermittent connectivity and highly-constrained bandwidth. Perhaps the most interesting and challenging requirement comes into play when policies must address issues that simultaneously affect multiple application domains.

In this paper we discuss the application of the KAoS policy services framework to human-robot teamwork—an application that involves a variety of application domains and enforcement at different levels of control; from low level network resource control to high level organizational constraints and coordination management. The study culminated in an outdoor field exercise that required coordination of mixed subteams composed of two people and five robots. The team communicated over standard wireless 802.11b but used and advanced network infrastructure. The mission was a hide-and-seek style task that involved searching for and apprehending an intruder on a Navy pier.

We will first briefly describe the KAoS framework and then will follow with a description of our human-robot teamwork application. We then discuss the range of policy semantics and enforcement strategies involved in this application.

2. KAoS Overview

The KAoS policy services framework [5] has been adapted to run on a variety of agent, robotic, Web services, Grid services, and traditional distributed computing platforms (e.g., [6][7][9][15]) and across a variety of industrial, military, and space applications. In addition to services directly related to policy management, KAoS also provides the basic services for distributed computing, including message transport and directory services. Because of the structure of the system as a set of services with the well-defined Common Services Interface (CSI), the application

developer can use all or any subset of its capabilities (e.g., registration, deconfliction, transport, publish-subscribe) as appropriate in a given situation. KAoS domain and policy services provide for the specification and management of roles and similar groupings, facilitating dynamic team formation and modification in human-automation applications.

KAoS supports two main types of policies. The set of permitted actions is determined by *authorization policies* that specify which actions an actor or set of actors is allowed (*positive authorizations* policies) or not allowed (*negative authorizations* policies) to perform in a given context. *Obligation policies* specify actions that an actor or set of actors is required to perform (*positive obligations*) or for which such a requirement is waived (*negative obligations*). All other kinds of policies (e.g., delegation, teamwork coordination) are constructed from these two primitive types, combined with other aspects of KAoS policy semantics (e.g., domains, history, state).

KAoS policies are expressed in OWL (Web Ontology Language: <http://www.w3.org/2004/OWL>), optionally augmented with other constructs (e.g., role-value maps) for greater expressivity. A graphical user interface, KPAT¹ (KAoS Policy Administration Tool), allows menu- or wizard-based policy construction with no need for understanding OWL. Additional details regarding KAoS are explained in [5][14].

3. Human-Robot Field Study

Our field study involved the problem of discovering and apprehending an intruder on a cluttered Navy pier using teams composed of a combination of two humans and five heterogeneous robots working in close and continuous interaction. The policies had to address a diversity of issues: network management, robotic behavior management, team modeling, and coordination. The humans use natural language to communicate with each other and with the robots [16]. This included the establishment of teams, assignment of leaders and tasking. The robots were capable of being controlled at any level from direct teleoperation to performing the search autonomously. The goal was to leverage the available team members and coordinate activity for a more effective search. Each of the seven team members communicated through an advanced network infrastructure called the Agile Computing Infrastructure (ACI) [10] that leverages new ideas to increase the efficiency and policy control of information flow in resource-constrained and dynamic mobile ad hoc networks [11].

¹ Pronounced “KAY-pat.”

4. Types of Polices

4.1. Network Management

KAoS policies have been used to provide help with various standard aspects of network management, such as access control, access prioritization, data filtering and transformation. For example, an access prioritization policy could provide a user that is teleoperating a robot a higher frame rate for video images than someone who is monitoring this sensor for surveillance. Filtering policies can involve, for example, removal of unauthorized information or delaying access to certain users. Polices can also be used to transform that data enroute, for example reducing the resolution of images to meet classification or bandwidth requirements.

Enforcement of access, bandwidth, transformation, or resource limitation polices is performed by the ACI, which can query the policy service to determine the constraints relevant to the current context.

4.2 Robotic Behavior Management

Critical constraints are often buried in the code of autonomous agents, making it difficult for anyone but the robot’s developer to understand, let alone manipulate them. Even when such constraints are represented declaratively, reducing barriers of understanding and runtime manipulation, management and deconfliction can be difficult. Finally, a lack of adequate semantics for expressing and reasoning about the context in which a particular constraint functions hampers the development of complex robot behaviors that work well in real-world situations.

Our KAoS-Robot component [8] provides a generic wrapper that both adapts to the capabilities of each robot and provides a consistent interface for client systems to access them. KAoS-Robot allows for control at various levels without restricting the control architecture. It also enables policy checking and enforcement for the robots, providing the increased awareness of robot state required to handle the regulation required for coordination. Using KAoS policies in conjunction with the KAoS-Robot component, we can do things like dynamically defining and enforcing areas of restricted movement or obligating the robot to perform certain actions (e.g., signaling, logging, requesting permission) in a given context.

4.3 Team modeling

Roles are a convenient aid in coordination and task allocation. They can be thought of as ways of packaging rights and obligations as policy sets that go

along with the necessary parts that participants play in joint activities. Collections of roles are often grouped to form higher level organizations.

Although in other studies we have experimented with teams of peers, in this study we implemented the role of “Leader” for each team. Leaders are subject to regulations pertaining to their own role(s), but may alter the pattern of activity and regulation in other team members.

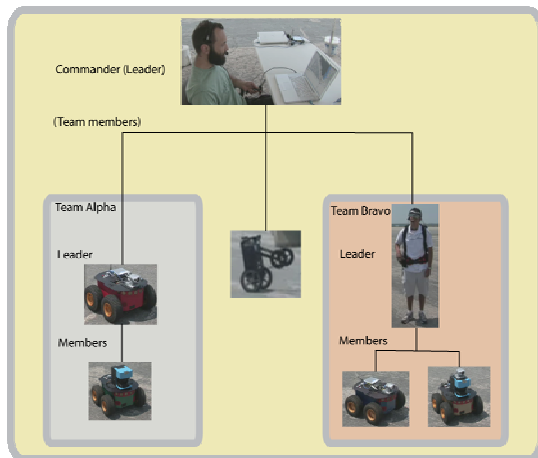


Figure 1 Hierarchical team structure composed of umbrella team and two sub teams.

In our field demonstration, the Commander, who has responsibility for the overall team, dynamically creates two sub teams of arbitrary composition and complexity, assigning leaders and a subset of potential team members to each using the KAoS Directory Service. The team structure is further modified at a later time. Agents can join and leave teams (or *be made to* join and leave) as necessary to support the desired structure. Roles can also be defined and adjusted dynamically.

4.4 Coordination

For the last several years, we have studied how humans and automation succeed and fail in joint activity requiring a high degree of interdependence among the participants [17]. Such interdependence requires that, in addition to what team members do to accomplish the work itself, they also invest time and attention in making sure that distributed or sequenced tasks are appropriately *coordinated*. Through the use of both highly-reusable generic and task-specific coordination policies and the offloading of many of the low-level coordination tasks to automated components, human attention can remain focused on critical tasks and costly errors can be avoided. Within this study, we experimented with several classes of such policies,

including acknowledging requests, notifying team members when appropriate, role management, and enforcing the chain of command [9].

4.5 Policy Management across Multiple Application Domains

Although each domain of application we address in our study is interesting in its own right, policies that cut across multiple domains are of special concern.

In our application, the network domain and autonomous agent domain are linked since the network is formed by the agents themselves. The ACI proactively moves resources as needed to support the requirements of the mobile ad hoc network. For example, if a robot moves out of range and loses communication, the ACI will recognize the failure and attempt to proactively restore communications by obligating the movement of an available resource that is authorized to serve as a network node [8]. The group of policies specified to implement this example uses context from both domains.

The ACI was also used to optimize bandwidth to the varying requirements of client roles by dynamic transformation of video streams from the robots [11][12]. This policy set exemplified interdependencies between the network and the team modeling domains.

Run-time bindings determined the resultant action that coupled the autonomous agent domain with the team modeling domain. For example, a policy to notify the supply clerk when using the last item involves resolving who is currently in the role of supply clerk. Similarly the coordination domain also directly affects the autonomous agent domain. A good example of this is the assignment of a leader to a team. While all team members are typically notified when a problem is encountered, some kinds of notification are best directed to the leader alone.

KAoS has a modular architecture so that various reasoning components can be plugged in for new services that extend the existing services and augment the capabilities of team members. One example is a component called Kaa (KAoS Adjustable Autonomy) that is built on the foundation of the current policy mechanisms by performing adjustments of autonomy consistent with policy [13][14]. Kaa monitors various aspects of team performance and can automatically or semi-automatically adjust autonomy along whichever dimensions (possibility, performability, authorization, obligation) are deemed to provide the highest team utility. Kaa is supported by another component, Kab (KAoS adaptive backup) which, in conjunction with Kaa, derives workarounds for recurrent classes of

failure conditions. For instance, when a robot is obliged to detect a motion but its camera (a main device for the role) fails, Kab could suggest using a different device (i.e., sonar) and/or using delegating the obligation to another robot. Using decision-theoretic algorithms, Kaa considers the costs and benefits of all feasible options (including suspension of the obligation), and determines the best course of action. The ability to specify and enforce policies across a range of domains in response to team changes, role adjustment, and network limitations makes all this possible.

5. Conclusion

Requirements for policy management systems to simultaneously support a diversity of domains that may include close and continuous human interaction are likely to become more common in the future. Our human-robot field study provided valuable experience in such an application and highlighted several interesting issues in this regard.

6. References

- [1] A.M. Hadjiantonis, A. Malatras, G. Pavlou, A context-aware, policy-based framework for the management of MANETs, Proc. of the Seventh IEEE International Workshop on Policies for Distributed Systems and Networks, May 2006 (IEEE POLICY 2006)
- [2] Lalana Kagal, "A Policy-Based Approach to Governing Autonomous Behavior in Distributed Environments", PhD Thesis, September 2004
- [3] Bacon, J., Lloyd, M., and Moody, K. 2001. *Translating role-based access control policy within context*. In Proceedings of the 2nd International Workshop on Policies for Distributed Systems and Networks (POLICY). Lecture Notes in Computer Science, 1995, Springer Verlag.
- [4] Yuri Demchenko, Leon Gommans, Cees de Laat, Andrew Tokmakoff, Rene van Buuren. Policy Based Access Control in Dynamic Grid-based Collaborative Environment. 2006 International Symposium on Collaborative Technologies and Systems (CTS2006)
- [5] Bradshaw, J.M., et al. (2003). Representation and reasoning for DAML-based policy and domain services in KAoS and Nomads. Proceedings of the Autonomous Agents and Multi-Agent Systems Conference (AAMAS 2003). Melbourne, Australia, New York, NY: ACM Press,
- [6] Bradshaw, J. M., et al. (2004). Making agents acceptable to people. In N. Zhong & J. Liu (Ed.), *Intelligent Technologies for Information Analysis: Advances in Agents, Data Mining, and Statistical Learning*. (pp. 355-400). Berlin: Springer Verlag.
- [7] Johnson, M., Chang, P., Jeffers, R., Bradshaw, J. M., Soo, V.-W., Breedy, M. R., Bunch, L., Kulkarni, S., Lott, J., Suri, N., & Uszok, A. (2003). KAoS semantic policy and domain services: An application of DAML to Web services-based grid architectures. Proceedings of the AAMAS 03 Workshop on Web Services and Agent-Based Engineering. Melbourne, Australia,
- [8] Johnson, M., Jeffrey M. Bradshaw, Paul Feltovich, Renia Jeffers, Hyuckchul Jung, and Andrzej Uszok, A Semantically Rich Policy Based Approach to Robot Control, Proceedings of the International Conference on Informatics in Control, Automation and Robotics, 2006
- [9] Johnson, M., Feltovich, P.J., Bradshaw, J.M., & Bunch, L. (in press, 2008). Human-Robot Coordination through Dynamic Regulation. In *Proceedings of the International Conference on Robotics and Automation (ICRA)*. Pasadena, California, May 19-23.
- [10] Suri, N., M. Carvalho, and J.M. Bradshaw. "Proactive resource management for agile computing." Presented at the Proceedings of the Tenth Annual ECOOP Workshop on Mobile Object Systems and Resource-Aware Computing, Oslo, Norway, 14 June, 2004.
- [11] Carvalho, M. "FlexFeed: A Mobile Agent-Based Framework for Dynamic Sensor Networks." Masters, University of West Florida, 2004.
- [12] Suri, N., J.M. Bradshaw, M.H. Burstein, A. Uszok, B. Benyo, M.R. Breedy, M. Carvalho, D. Diller, P.T. Groth, R. Jeffers, M. Johnson, S. Kulkarni, and J. Lott. "DAML-based policy enforcement for semantic data transformation and filtering in multi-agent systems." Presented at the Proceedings of the Autonomous Agents and Multi-Agent Systems Conference (AAMAS 2003), Melbourne, Australia, 14-18 July, 2003.
- [13] Bradshaw, J.M., P. Feltovich, H. Jung, S. Kulkarni, W. Taysom, and A. Uszok. "Dimensions of adjustable autonomy and mixed-initiative interaction." In *Agents and Computational Autonomy: Potential, Risks, and Solutions. Lecture Notes in Computer Science, Vol. 2969*, edited by M. Nickles, M. Rovatsos, and G. Weiss, 17-39. Berlin, Germany: Springer-Verlag, 2004.
- [14] Bradshaw, J.M., H. Jung, S. Kulkarni, M. Johnson, P. Feltovich, J. Allen, L. Bunch, N. Chambers, L. Galescu, R. Jeffers, N. Suri, W. Taysom, and A. Uszok. "Toward trustworthy adjustable autonomy in KAoS." In *Trusting Agents for Trustworthy Electronic Societies*, edited by R. Falcone. Berlin: Springer, 2005.
- [15] Uszok, A., J.M. Bradshaw, M. Johnson, R. Jeffers, A. Tate, J. Dalton, and S. Aitken. "KAoS policy management for semantic web services." *IEEE Intelligent Systems* 19, no. 4 (July/August 2004): 32-41.
- [16] Allen, J., N. Blaylock, and G. Ferguson. "A problem-solving model for collaborative agents." Proceedings of the International Joint Conference on Autonomous Agents and Multi-Agent Systems (AAMAS 2002), 2002.
- [17] Klein, G., D.D. Woods, J.M. Bradshaw, R. Hoffman, and P. Feltovich. "Ten challenges for making automation a "team player" in joint human-agent activity." *IEEE Intelligent Systems* 19, no. 6 (November-December 2004): 91-95.